Large-Scale Complex Adaptive Systems using Multi-Agent Modeling and Simulation

(Extended Abstract)

Lachlan Birdsey, Claudia Szabo, Katrina Falkner
School of Computer Science
The University of Adelaide
Adelaide, Australia
{lachlan.birdsey, claudia.szabo, katrina.falkner}@adelaide.edu.au

ABSTRACT

Modeling and analysis of large-scale complex adaptive systems (CAS) is critical to understanding their key properties such as self-organization, emergence, and adaptability. These properties are difficult to analyze in real-world scenarios due to performance constraints, metric design, and limitations in existing modeling tools. In our previous work, we proposed the Complex Adaptive Systems Language (CASL) and its associated framework. In this paper, we introduce CASL-SG, a *Semantic Group* extension for large-scale modeling using relational hierarchies. CASL-SG permits large-scale simulations to be executed on modest hardware by enabling simulations to contain approximately twice as many agents. This achieves a 356% runtime improvement for a Game of Life model with 4 million cells. CASL-SG also enables designers to model behaviors of collectives that drive the system and entities towards self-organization and adaptability.

Keywords

Agent-Based Simulation; Complex Adaptive Systems

1. INTRODUCTION

Complex adaptive systems (CAS) are attracting significant research interest in domains such as social networks, supply chains, health-care networks, smart-cities and smart-grids, the 'Internet of Things', and the Internet [2, 9, 15, 17], where modeling of known systems as CAS provides valuable insight into behaviors, communication patterns, and the effect of various policies. CAS are a type of complex system where entities and the environment adapt and interact in order to achieve desired properties [10, 13, 18] and provide a realistic abstraction of real-life scenarios [15, 17]. At the same time, due to their many autonomous and interconnected entities, unexpected and emergent properties can appear [5, 13, 20]. In CAS modeling, the importance of entity interaction, adaptation, and influence from the operating environment is highlighted, enabling deeper study focusing on individuals and how their behaviors contribute to higher level properties [10, 13, 18].

The study of CAS faces many modeling and implementation challenges. It is important that CAS modeling tools are not domain specific [11, 16, 17] and also allow for the modeling and subsequent investigation of key CAS properties such as adaptability [6, 14, 21] and self-organization [7, 8, 10, 13]. Domain agnostic modeling approaches tend to use generic paradigms such as agent-based modeling but suffer from issues of scale [4], and do not provide a solid grounding to model individual agent and environment adaptations. Furthermore, they can only consider a single form of agent representation, such as a network or GIS, which hinders attempts at creating highly detailed realistic CAS models. We resolve these issues by extending CASL [3] by introducing a new class of entity called a '*semantic group*'.

2. CASTLE SIMULATION FRAMEWORK

Our prototype framework, called CASTLE, consists of the CASL modeler, the code generator, a simulator, and the observation tool. Once a model is constructed in the CASL modeler, code is generated only when all the required constraints have been adhered to. The generated code is then executed in the simulator, which may require initialization parameters that can be provided by a configuration file. The observation tool is comprised of several modules that analyze various features of a CAS such as aggregation, runtimes, interactions, and domain-specific features. It is designed to be extensible to allow for new metrics to be added, either designed specifically for the current simulation or for a more domain-agnostic purpose. CASTLE is available online[1].

2.1 Achieving Scale with Semantic Groups

We extend CASL and initial framework by considering the collective relationship between entities. Each collective, or group, consists of agents that have a semantic relationship. We refer to these collectives as '*semantic groups*'. The relationship that forms the basis of the semantic groups is dependent on how particular agents are represented and in many cases, how agents are represented can form the basis for their relationship. For example, in an Emergency Department, a patient has a stronger relationship with a doctor or a nurse, than a pathology technician. Semantic groups can be considered similar to 'swarms' in SWARM [12], however semantic groups allow for different representations across

[1] `https://github.com/CASTLE-FWK`

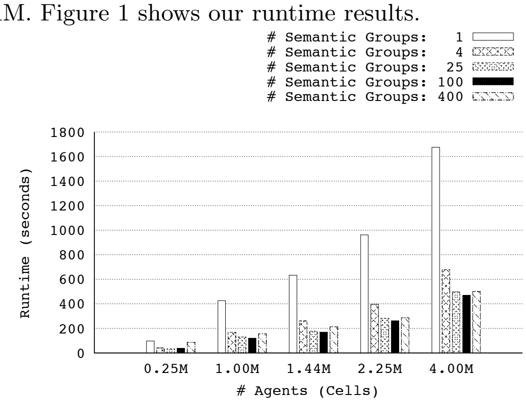
Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

multiple groups in the same system, and permit automated and constantly active parallelization. Semantic groups also encourage the modeling of self-organization and adaptability. By abstracting model components as those with collective behaviors, the focus can be on the way these collectives interact, adapt, and create self-organized structures. Furthermore, as semantic groups possess their own states and behaviors, they can be the main drivers for the system as a whole to undergo adaptation, and also provide assistant functions for the agents contained within. For example, in a Flock of Birds model [19], semantic groups could contain previously flocked Boids, to examine how flocks comprised of different Boids behave and interact in the same environment. As these groups of entities are established at design-time by the modeler, code suitable for parallel simulation can be generated. Environments run in parallel with other environments, and semantic groups run in parallel with other semantic groups. Agents contained in semantic groups do not run in parallel, but instead have their execution order shuffled, similar to normal agent-based simulation. At simulation time, the user specifies the number and types of semantic groups required, similar to other entities. At simulation time, each semantic group is assigned a thread, process, or node, which is controlled by the entity that contains the semantic group.

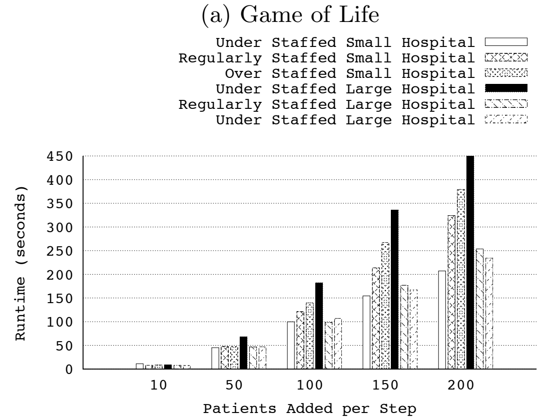
To facilitate semantic groups in CASL, we introduce a new entity type called `SEMANTIC_GROUP`, which contains the same concerns as the other entities except that `INTERACTIONS` are now replaced by `INTERNAL_INTERACTIONS` and `EXTERNAL_INTERACTIONS`. We separate the interactions for three reasons. Firstly, semantic groups are a separate entity type, requiring extra features for implementation and code generation, such as specific rules on how groups interact. Secondly, some interactions involving groups may serve a maintenance-like purpose such as information transfer or synchronization. Finally, the separation allows semantic groups to conform to the rules of a CASL simulation step. At simulation time, each step is broken into three phases, namely, *Setup*, *Action*, and *Cleanup*. The *Setup* and *Cleanup* phases allow for messages to be passed between groups and environments, as well as the execution of various agent behaviors. This allows for agents contained in one semantic group to be able to interact with agents located in a different semantic group. The *Action* phase is where the bulk of low-level entity interactions occur; no interactions between groups or environments are allowed. We restrict the inter-group and inter-environment interactions to the *Setup* and *Cleanup* phases, as the individual agents behaviors and interactions are more computationally intensive and thus large transfers between groups and environments are isolated. Each entity contains separate execution queues for each phase, which allows actions to be queued and triggered at the correct time. As there is a large emphasis on the design of modular agents, semantic groups, and environments, CASL maintains specific rules for what is allowed to occur during step phases. For example, transmissions between semantic groups use the `EXTERNAL_INTERACTIONS` rule which can only be triggered in an *Setup* or *Cleanup* phase.

3. RUNTIME ANALYSIS

We implement two models using CASL-SG, namely, the Game of Life and an Emergency Department model based on Australian Health Department guidelines [1]. For both models we run several experiments to highlight the scalability of CASL-SG. For the Game of Life model, we run simulations for 1,000 steps with increasing `Cell` populations (up to 4 million) and compare runtimes when there are 1, 4, 25, 100, or 400 semantic groups. For the Emergency Department model, we run simulations for 4,380 steps, increasing the number of `Patient`s introduced per hour. Each simulation is based on 6 increasingly larger scenarios from an *Under Staffed Small Hospital* with 20 `Nurses` in 1 `WaitingRoom`, 8 `Doctors` in 1 `DoctorClinic`, 10 `PathologyTechnician` in 1 `PathologyLab`, increasing to an *Over Staffed Large Hospital* with 100 `Nurses` across 10 `WaitingRooms`, 50 `Doctors` across 10 `DoctorClinics`, 50 `PathologyTechnician` across 10 `PathologyLab`s. Each runtime experiment is replicated 10 times and the mean runtime is presented. All experiments have been executed on a commodity laptop, with a 2.8GHz Intel i7 CPU with the JVM being limited to 8GB of RAM. Figure 1 shows our runtime results.



(a) Game of Life



(b) Emergency Department

Figure 1: Experiment Runtimes

4. CONCLUSION

Large-scale CAS modeling and simulation provide significant challenges as they contain a vast number of complex, interacting entities, in particular being able to efficiently design a model and be able to execute simulations with few performance concerns. In this paper we propose a semantic group modeling language supported by a simulation and analysis framework. We show that when a CASL model utilizes semantic groups, the simulation scales well, with a Game of Life simulation with 4 million agents taking 28 minutes with 1 semantic group, reduced to 8 minutes with 100 semantic groups. Similarly, our largest Emergency Department simulation took 7.5 minutes with 3 semantic groups, which reduced to 3.5 minutes with 30 semantic groups.

REFERENCES

[1] Australian Government. Department of Health | Emergency Triage Education Kit. `http://www.health.gov.au/internet/main/publishing.nsf/Content/casemix-ED-Triage+Review+Fact+Sheet+Documents`. Last Accessed: 01/05/2016.

[2] M. Benham-Hutchins and T. R. Clancy. Social Networks As Embedded Complex Adaptive Systems. *Journal of Nursing Administration*, 40(9):352–356, 2010.

[3] L. Birdsey, C. Szabo, and K. Falkner. CASL: A Declarative Domain Specific Language For Modeling Complex Adaptive Systems. In *Proceedings of the Winter Simulation Conference*, 2016.

[4] F. Boulaire, M. Utting, and R. Drogemuller. Dynamic Agent Composition For Large-Scale Agent-Based Models. *Complex Adaptive Systems Modeling*, 3(1):1–23, 2015.

[5] W. K. V. Chan, Y. Son, and C. Macal. Agent-based Simulation Tutorial - Simulation Of Emergent Behavior And Differences Between Agent-based Simulation And Discrete-Event Simulation. In *Proceedings of the Winter Simulation Conference*, pages 135–150, 2010.

[6] C. Chira, A. Gog, R. I. Lung, and D. Iclanzan. Complex Systems And Cellular Automata Models in the Study of Complexity. *Studia Informatica Series*, 55(4):33–49, 2010.

[7] R. Dekkers. Complex Adaptive Systems. In *Applied Systems Theory*, pages 169–190. Springer International Publishing Switzerland, 2015.

[8] A.-M. Grisogono. Co-adaptation. In *Microelectronics, MEMS, and Nanotechnology*, pages 603903/1–603903/15. International Society for Optics and Photonics, 2005.

[9] A. Hasgall. Digital Social Networks As Complex Adaptive Systems. *Vine*, 43(1):78–95, 2013.

[10] J. H. Holland. Studying Complex Adaptive Systems. *Journal of Systems Science and Complexity*, 19(1):1–8, 2006.

[11] J. H. Miller and S. E. Page. *Complex Adaptive Systems: An Introduction To Computational Models Of Social Life*. Princeton University Press, 2009.

[12] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations. Technical report, Swarm Development Group, 1996.

[13] S. Mittal. Emergence In Stigmergic And Complex Adaptive Systems: A Formal Discrete Event Systems Perspective. *Cognitive Systems Research*, 21:22–39, 2013.

[14] G. E. Mobus and M. C. Kalton. *Principles of Systems Science*. Springer, 2015.

[15] M. A. Niazi. Complex Adaptive Systems Modeling: A Multidisciplinary Roadmap. *Complex Adaptive Systems Modeling*, 1(1):1–14, 2013.

[16] M. A. Niazi and A. Hussain. *Cognitive Agent-Based Computing-I: A Unified Framework for Modeling Complex Adaptive Systems Using Agent-Based & Complex Network-based Methods*, volume 1 of *SpringerBriefs in Cognitive Computation*. Springer Netherlands, 2013.

[17] M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. Complex Adaptive Systems Modeling With Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1):3, 2013.

[18] Ö. Özmen, J. Smith, and L. Yilmaz. An Agent-based Simulation Study Of A Complex Adaptive Collaboration Network. In *Proceedings of the Winter Simulation Conference*, pages 412–423, 2013.

[19] C. Reynolds. Flocks, Herds And Schools: A Distributed Behavioral Model. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 25–34, 1987.

[20] C. Szabo, Y. M. Teo, and G. K. Chengleput. Understanding Complex Systems: Using Interaction As A Measure Of Emergence. In *Proceedings of the Winter Simulation Conference*, pages 207–218, 2014.

[21] B. Walker, C. S. Holling, S. R. Carpenter, and A. Kinzig. Resilience, Adaptability and Transformability in Social–Ecological Systems. *Ecology and Society*, 9(2):5, 2004.